Faculté des Sciences & Techniques — Université de Limoges

Master 2nde année

TMC

# Pico C&ASM

*Mélange C & Assembleur*

## Déploiement du projet

Vous trouverez le contenu du projet MPU sur :
Dans le projet, vous aurez un script pour préparer automatiquement `renode` :

`launch_in_renode.resc`

```
include @boards/initialize_raspberry_pico.resc
sysbus LoadELF @/BUILD/buildpico-uart/asm_c_interop.elf
cpu0 PC 0x100001E9
cpu1 PC 0x100001E9
machine StartGdbServer 3333
showAnalyzer sysbus.uart0
```

Pour fabriquer le firmware :

xterm
```
$ export PICO_SDK_PATH=~/pico-sdk
$ make renode-rp2040
```

Vous lancerez la simulation par :

xterm
```
$ docker run -it --rm --name renode -p 8080:8080 -p 3333:3333 -v $(pwd):/BUILD renode-rp2040
21:12:12.0514 [WARNING] Couldn't start UI - falling back to console mode
21:12:12.3183 [INFO] Loaded monitor commands from: /opt/renode/scripts/monitor.py
Renode, version 1.16.1 (d66b0c2a-202602160923)

(monitor) include @/BUILD/launch_in_renode.resc
```

*Le `include` lance le script à exécuter par renode : il charge le firmware et configure son exécution.*

Ensuite vous pourrez vous connecter avec `gdb` :

xterm
```
$ gdb-multiarch -ex "target extended-remote :3333" build-pico-uart/asm_c_interop.elf
```

## Analyse de la combinaison de fonction C et ASM

**1 –** Vous décrirez la structure et le fonctionnement du firmware :
a. suivant le linker ;
b. en étudiant la combinaison de fonctions C et assembleur.

**2 –** a. Qu'est-ce qu'une fonction « naked » ?
b. Que veut dire `.thumb_func` ?
c. À quoi servent les `push` et `pop` ?
d. Comment sont gérés les « *symbols* » ?
e. Comment marche le passage de paramètres ?
f. Où se trouve la valeur de retour d'une fonction ?
g. Comment sont créées les variables locales à une fonction ?
h. Comment fonctionne la pile ?
i. Qu'est-ce qu'une fonction « *leaf* » ? Quel rapport avec le registe `lr` ?
j. Quels sont les « *Pitfalls* » ?